Artificial Intelligence Lab
AL-2002

Lab 03

Instructor: Hurmat Hidayat

Semester: Spring 2023

Artificial Intelligence Lab 03

Objectives

The objective of this AI lab is to provide students with a comprehensive understanding of supervised machine learning, specifically the K-Nearest Neighbor (KNN) algorithm, and its applications in classification.

Learning Outcomes

1. Implement the K-Nearest Neighbor algorithm for solving classification problems.
2. Evaluate the advantages and disadvantages of using the KNN algorithm.
3. Apply the KNN algorithm to real-world datasets and analyze the results.

Table of Contents

Machine Learning

Machine learning is subtype of Artificial Intelligence. Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed. ML solves problems that cannot be solved by numerical means alone.

"*A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.*" -- *Tom Mitchell, Carnegie Mellon University*

In the various problem settings T, P, and E can refer to completely different things. Some of the most popular tasks T in machine learning are the following:

- classification of an instance to one of the categories based on its features;
- regression – prediction of a numerical target feature based on other features of an instance;
- clustering – identifying partitions of instances based on the features of these instances so that the members within the groups are more similar to each other than those in the other groups;
- anomaly detection – search for instances that are "greatly dissimilar" to the rest of the sample or to some group of instances;
- and so many more.

Experience E refers to data (we can't go anywhere without it). Machine learning algorithms can be divided into those that are trained in supervised or unsupervised manner.

Supervised machine learning: The program is "trained" on a pre-defined set of "training examples" with given class labels, which then facilitate its ability to reach an accurate conclusion when given new data.

Unsupervised machine learning

The program is given a bunch of data and must find patterns and relationships therein.

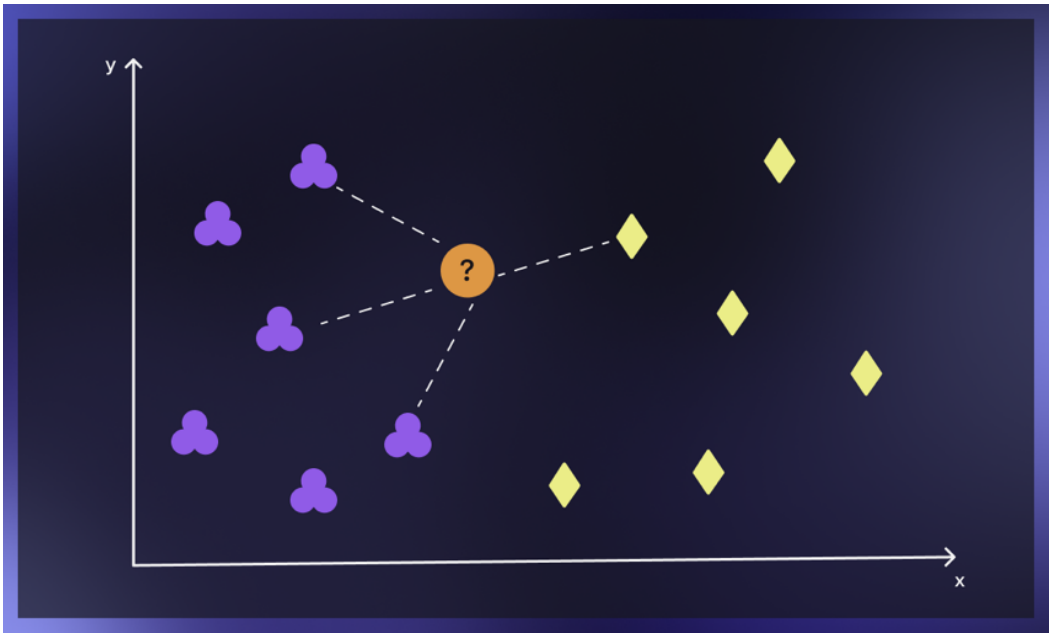
Supervised Learning (Classification)

K-Nearest Neighbor

The k-nearest neighbors classifier (kNN) is a non-parametric supervised machine learning algorithm. It's distance-based: it classifies objects based on their proximate neighbors' classes. kNN is most often used for classification, but can be applied to regression problems as well.

Non-parametric means that there is no fine-tuning of parameters in the training step of the model. Although k can be considered an algorithm parameter in some sense, it's actually a hyperparameter. It's selected manually and remains fixed at both training and inference time.

The k-nearest neighbors algorithm is also **non-linear**. In contrast to simpler models like linear regression, it will work well with data in which the relationship between the independent variable (x) and the dependent variable (y) is not a straight line.



What is k in k-nearest neighbors?

The parameter k in kNN refers to the number of labeled points (neighbors) considered for classification. The value of k indicates the number of these points used to determine the result. Our task is to calculate the distance and identify which categories are closest to our unknown entity.

K-Nearest Neighbors Algorithm

Here is step by step on how to compute K-nearest neighbors KNN algorithm:

1. Determine parameter K = number of nearest neighbors. "**K" should be an Odd**, it helps in picking majority votes. If K=4 => 2 rows have label '0" and 2 rows have label "1", so it is very difficult to pick majority label.
2. Calculate the distance between the query-instance and all the training samples
3. Sort the distance and determine nearest neighbors based on the K-th minimum distance
4. Gather the Y (labels) of only nearest neighbors. Use simple majority of the Y (labels) of nearest neighbors as the prediction value of the query instance

Case Study

We have data from the questionnaires survey (to ask people opinion) and objective testing with two attributes (acid durability and strength) to classify whether a special paper tissue is good or not. Here is four training samples.

X1 = Acid Durability	X2 = Strength	Y = Classification
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

Now the factory produces a new paper tissue that pass laboratory test with **X1 = 3 and X2 = 7**.

Without another expensive survey, can we guess what the classification of this new tissue is?

1. Determine parameter K = number of nearest neighbors; Suppose use K = 3
2. Calculate the distance between the query-instance and all the training samples
    a. Coordinate of query instance is (3, 7), instead of calculating the distance we compute square distance which is faster to calculate (without square root)

X1 = Acid Durability	X2 = Strength	Euclidian Distance with query (3,7)
7	7	$(7-3)^2 + (7-7)^2 = 16$
7	4	$(7-3)^2 + (4-7)^2 = 25$
3	4	$(3-3)^2 + (4-7)^2 = 9$
1	4	$(1-3)^2 + (4-7)^2 = 13$

3. Sort the distance and determine nearest neighbors based on the K-th minimum distance

X1 = Acid Durability	X2 = Strength	Euclidian Distance with query (3,7)	Rank Min. Distance	Included
7	7	$(7-3)^2+(7-7)^2=16$	3	Yes
7	**4**	$\mathbf{(7-3)^2+(4-7)^2=25}$	**4**	**No**
3	4	$(3-3)^2+(4-7)^2=9$	1	Yes
1	4	$(1-3)^2+(4-7)^2=13$	2	Yes

4. Gather the category of the nearest neighbors. Notice in the second row last column that the category of nearest neighbor (Y) is not included because the rank of this data is more than 3 (=K).

X1 = Acid Durability	X2 = Strength	Euclidian Distance with query (3,7)	Rank Min. Distance	Included	Y = Label
7	7	$(7-3)^2+(7-7)^2=16$	3	Yes	Bad
7	**4**	$\mathbf{(7-3)^2+(4-7)^2=25}$	**4**	**No**	**-**
3	4	$(3-3)^2+(4-7)^2=9$	1	Yes	Good
1	4	$(1-3)^2+(4-7)^2=13$	2	Yes	Good

5. Use simple majority of the category of nearest neighbors as the prediction value of the query instance. We have 2 good and 1 bad, since 2 > 1, then we conclude that a new paper tissue that pass laboratory test with X1 = 3 and X2 = 7 is included in Good category.

Advantages

- **No Training** - KNN is called Lazy Learner (Instance based learning). It does not learn anything in the training period. It does not derive any discriminative function from the training data. In other words, there is no training period for it. It stores the training dataset and learns from it only at the time of making real time predictions. This makes the KNN algorithm much faster than other algorithms that require training e.g. SVM, Linear Regression etc.
- **New Data** - Since the KNN algorithm requires no training before making predictions, new data can be added seamlessly which will not impact the accuracy of the algorithm.
- **Easy Implementation** - KNN is very easy to implement. There are only two parameters required to implement KNN i.e. the value of K and the distance function (e.g. Euclidean or Manhattan etc.)

- Variety of distance criteria to be choose from: K-NN algorithm gives user the flexibility to choose distance while building K-NN model.
    - Euclidean Distance
    - Hamming Distance
    - Manhattan Distance
    - Minkowski Distance

Disadvantages

- **Large Datsets** - Does not work well with large dataset: In large datasets, the cost of calculating the distance between the new point and each existing points is huge which degrades the performance of the algorithm
- **High Dimensionality Problems** - KNN performs best with a low number of features. When the number of features increases, then it requires more data. When there's more data, it creates an overfitting problem.
- **Feature Scaling** - Need feature scaling: We need to do feature scaling (standardization and normalization) before applying KNN algorithm to any dataset. If we don't do so, KNN may generate wrong predictions
- **Sensitive** - Sensitive to noisy data, missing values and outliers: KNN is sensitive to noise in the dataset. We need to manually impute missing values and remove outliers.

Practical applications

1. K-nearest neighbors method for car manufacturing

An automaker has designed prototypes of a new truck and sedan. To determine their chances of success, the company has to find out which current vehicles on the market are most similar to the prototypes. Their competitors are their "nearest neighbors." To identify them, the car manufacturer needs to input data such as price, horsepower, engine size, wheelbase, curb weight, fuel tank capacity, etc., and compare the existing models. The kNN algorithm classifies complicated multi-featured prototypes according to their closeness to similar competitors' products.

2. kNN in E-commerce

K-nearest neighbors is an excellent solution for cold-starting an online store recommendation system, but with the growth of the dataset more advanced techniques are usually needed. The algorithm can select the items that specific customers would like or predict their actions based

on customer behavior data. For example, kNN will quickly tell whether or not a new visitor will likely carry out a transaction.

3. kNN application for education

Another kNN application is classifying groups of students based on their behavior and class attendance. With the help of the k-nearest neighbors analysis, it is possible to identify students who are likely to drop out or fail early. These insights would allow educators and course managers to take timely measures to motivate and help them master the material.

Lab Tasks

Task 1. Develop the K-Nearest Neighbors (KNN) algorithm from scratch using Python, without relying on any libraries.

Task 2. Utilizing the fruit_data_with_colors.csv dataset, perform the following steps:

- **a.** Read and load the data into the program.
- **b.** Prepare the data by eliminating any features that contain text or categorical values.
- **c.** Address missing values by replacing them with the mean value of each column, if necessary.
- **d.** Divide the data into training and testing sets, with the first 50 rows being used for training and the remaining 10 rows being used for testing.
- **e.** Apply the KNN model for different values of K (ranging from 1 to 10) and examine the results.
- **f.** Plot the accuracy score for each value of K, to visualize the differences.

Home Activity:

Read the provided **Heart Failure Classification** with KNN and Decision Tree notebook thoroughly and carefully. Ensure that you understand each step and the purpose behind it, including the mathematical concepts and code implementation. If any step is unclear, research additional resources or reach out to the instructor for clarification.